

## **REMARKS**

Claims 1-10 and 12-21 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

### **Claim Objections**

The Examiner objected to claims 12-16 for “potential 101 problem” because the claims are directed to a process. Applicants respectfully traverse this objection. However, to expedite prosecution, Applicants have amended claims 12-16 to recite “a computer implemented method” as suggested by the Examiner. Therefore, removal of the objection to claims 12-16 is respectfully requested.

### **Section 101 Rejection:**

The Examiner rejected claim 11 under 35 U.S.C. § 101 as allegedly being directed to non-statutory subject matter. While Applicants traverse this rejection, claim 11 has been cancelled rendering this rejection moot.

### **Section 103(a) Rejections:**

The Examiner rejected claims 1, 5, 6, 10-12, 16, 17 and 21 under 35 U.S.C. § 103(a) as being unpatentable over Gigliotti et al. (U.S. Patent 6,393,458) (hereinafter “Gigliotti”). Applicants respectfully traverse these rejections for at least the following reasons.

In regard to claim 1, contrary to the Examiner’s assertion, the cited art does not teach or suggest *one or more client machines each configured to implement one or more clients of the application server, wherein each client on a respective one of the one or more client machines is configured to create a plurality of client-side Object Request Brokers (ORBs) on the client machine, wherein each client-side ORB is coupled to a*

*server-side ORB of a different one of the plurality of application server instances.* The Examiner equates client objects 45, 46, 48 and 50 of Gigliotti to the one or more clients recited in Applicants' claim 1. However, client objects 45, 46, 48 and 50 of Gigliotti do not each create a plurality of client-side Object Request Brokers (ORBs) on the client machine(s), wherein each client-side ORB is coupled to a server-side ORB of a different one of the plurality of application server instances, as recited in Applicants' claim 1. Gigliotti is completely silent on how any client-side ORBs are created. Gigliotti certainly does not suggest that each one of client objects 45, 46, 48 and 50 create a plurality of client-side Object Request Brokers (ORBs) on the client machine(s). In fact, Gigliotti states just the opposite. At col. 8, lines 3-6, Gigliotti specifically states that a client uses only a single ORB ("via an ORB") (emphasis added).

In the response to argument section of the Action dated November 26, 2008, in response to the above arguments, the Examiner asserts "In Gigliotti, client objects are implemented on a client machine. In addition, each client object is associated with a load balancer. As such, a client object and its associated load balancer can be interpreted to read upon the 'client' of Applicants' invention...it is entirely possible for each client object to have its own load balancer (col. 6, lines 11-17). Interpreted in this manner, it is inherent that the client object and its associated load balancer create a plurality of client-side object request brokers (ORBs) on the client machine."

The Examiner's assertion that it is "inherent that the client object and its associated load balancer create a plurality of client-side object request brokers (ORBs)" does not follow from the Examiner's argument that "a client object and its associated load balancer can be interpreted to read upon the 'client'", is pure speculation on the part of the Examiner, and is not supported by any evidence of record. Gigliotti is completely silent as to how any client-side ORBs are created. Gigliotti does not teach or even suggest that each one of the client objects create a plurality of client-side ORBs on the client machine(s). Even considering the Examiner's contention that "a client object and its associated load balancer can be interpreted to read upon the 'client'", Gigliotti does not teach or even suggest that each one of "clients" create a plurality of client-side ORBs

on the client machine(s).

Further in regard to claim 1, contrary to the Examiner's assertion, the cited art does not teach or suggest that each client on a respective one of the one or more client machines is configured to select one of the plurality of client-side ORBs created by that client on the client machine according to a load balancing scheme in response to a request to access the application server. In regard to this limitation, the Examiner cites col. 6, lines 37-39, of Gigliotti which merely states: "In an exemplary embodiment, a load balancer object determines a balanced distribution for events which have been published or initiated by the client." First of all, this is referring to the operation of the load balancer, not the client. The client objects 45, 46, 48 and 50 in Gigliotti do not perform any selection from among a plurality of client-side ORBs. In fact, as shown above, each client in Gigliotti uses only a single ORB.

In the response to argument section of the Action dated November 26, 2008, in response to the above arguments, the Examiner "disagrees for the same reasons as above. The client and the load balancer are equivalent to Applicants' client." Even considering the Examiner's contention that "the client and the load balancer are equivalent to Applicants' client", Gigliotti does not teach or even suggest that each one of "clients" create a plurality of client-side ORBs on the client machine(s), nor does Gigliotti teach or suggest that each client on a respective one of the one or more client machines is configured to select one of the plurality of client-side ORBs created by that client on the client machine according to a load balancing scheme in response to a request to access the application server.

Furthermore, to determine "a balanced distribution for events which have been published or initiated by the client," the load balancer in Gigliotti is not described as selecting one of the plurality of pre-existing client-side ORBs created by that client on the client machine. There is no mention whatsoever in Gigliotti of the load balancer selecting one of a plurality of pre-existing client-side ORBs created by the requesting client. While the load balancer in Gigliotti does select a server host to send an event

published by a client, Gigliotti does not describe that the load balancer selects among pre-existing client-side ORBs created by that client on the client machine. Even if the load balancer in Gigliotti used an ORB to send the event to the server host, the ORB may be dynamically obtained or created at that time. There is no description in Gigliotti of selecting among pre-existing client-side ORBs. Moreover, there is certainly no suggestion in Gigliotti of selecting among a plurality of pre-existing client-side ORBs created by the particular client that initiated the event (request).

In the response to argument section of the Action dated November 26, 2008, in response to the above arguments, the Examiner asserts “Gigliotti clearly uses ORB to provide a communication means to the host (col. 7, lines 59-60). Therefore, by choosing a host to route a request, the load balancer is essentially choosing an ORB through which to route the request. Gigliotti is clear that the load balancer is in constant communication with the host (col. 6, lines 17-21). Therefore, it is not necessary that the ORB is dynamically obtained or created at the time that the load balancer chooses a server host, as a communication means (i.e., ORB) already exists.” While the load balancer in Gigliotti does select a server host to send an event published by a client, Gigliotti does not teach or even suggest that the load balancer selects among pre-existing client-side ORBs created by that client on the client machine. There is no mention at all in Gigliotti of selecting among pre-existing client-side ORBs. Even if true that “by choosing a host to route a request, the load balancer is essentially choosing an ORB through which to route the request”, as the Examiner contends, this does not teach the limitations as recited in Applicants’ claim 1.

The Examiner states: “it would have been obvious to one of ordinary skill in the art, that ORBs exist at both the client and server in order for the ORB protocol to be used.” However, even if ORBs existed at both the client and server in Gigliotti, that would still not result in the specific limitations recited in Applicants’ claim 1. For example, even if ORBs existed at both the client and server in Gigliotti, that would not mean that each client in Gigliotti creates a plurality of client-side Object Request Brokers (ORBs) on the client machine, wherein each client-side ORB is coupled to a server-side

ORB of a different one of the plurality of application server instances, as required by Applicants' claim 1. Nor would it mean that each client in Gigliotti selects one of the plurality of client-side ORBs created by that client on the client machine according to a load balancing scheme in response to a request to access the application server. Thus, a *prima facie* rejection has not been established.

In the response to argument section of the Action dated November 26, 2008, in response to the above arguments, the Examiner simply asserts "Gigliotti teaches the claimed invention for the reasons argued above." The Examiner's arguments have failed to establish that Gigliotti teaches the limitations as recited in Applicants' claim 1. The Examiner has relied on speculation, interpretations, and arguments that are not supported by the actual teachings of Gigliotti. Gigliotti fails to teach what is recited in claim 1 when viewed as a whole, as Applicants have made clear above.

Thus, for at least the reasons presented above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 1 apply to claims 6, 11, 12, and 17.

The Examiner rejected claims 2-4, 7-9, 13-15 and 18-20 under 35 U.S.C. § 103(a) as being unpatentable over Gigliotti in view of AAPA. Applicants respectfully traverse these rejections for at least the following reasons.

In regard to claim 3, contrary to the Examiner's assertion, Gigliotti and AAPA does not teach or suggest that said creation of a plurality of client-side ORBs and said selection of one of the plurality of client-side ORBs according to a load balancing scheme are performed by a Context Factory class. The Examiner refers to the admitted existence of JNDI and states that it would have been obvious to modify Gigliotti to include the use of JNDI. However, merely using JNDI in Gigliotti would not result in the specific limitations recited in claim 3. Therefore, the Examiner has failed to state a *prima facie* rejection. More specifically, employing JNDI in Gigliotti would not mean that the clients in Gigliotti would use a Context Factory class to both create and select among a plurality

of client-side ORBs. There is absolutely no evidence of record whatsoever to support the rejection of this claim.

In the response to argument section of the Action dated November 26, 2008, in response to the above arguments, the Examiner asserts “AAPA is clear the JNDI allows for directory services in distributed object systems such as CORBA (p. 3 lines 6-14). Furthermore, the use of JNDI involves the use of interfaces such as the `javax.naming.Context` interface (p. 3, lines 16-24). It is also well known in the art to use the `InitialContextFactory` interface in the `javax.naming.spi` since an initial context must be created using the `InitialContextFactory` class...” The Examiner refers to an html document (`InitialContextFactory.html`) on the website `java.sun.com`.

Again, merely using JNDI in Gigliotti would not result in the specific limitations as recited in claim 3. More specifically, employing JNDI in Gigliotti would not mean that the clients in Gigliotti would use a Context Factory class to both create and select among a plurality of client-side ORBs. The `InitialContextFactory.html` document does not provide support for the Examiner’s assertions, nor does it overcome Applicants’ arguments, in regard to claim 3. There is no evidence of record to support the rejection of this claim. Therefore, the Examiner has failed to state a *prima facie* rejection.

Thus, for at least the reasons presented above, the rejection of claim 3 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 3 apply to claims 8, 14, and 19.

## CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-71800/RCK.

Respectfully submitted,

/Robert C. Kowert/

Robert C. Kowert, Reg. #39,255  
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8850

Date: January 29, 2009